Tutorial for Postman

This tutorial walks through calling a web API using the Postman tool.

Step One: Start the API server

First, run the command `npm install` to install any dependencies.

Then, start the project in VS Code by running `npm run serve`. This starts a local web API server that you'll use with Postman. The server is an API that simulates an Instagram-like API with users, images, and comments.

Once the server starts, you'll see a message that looks like this:

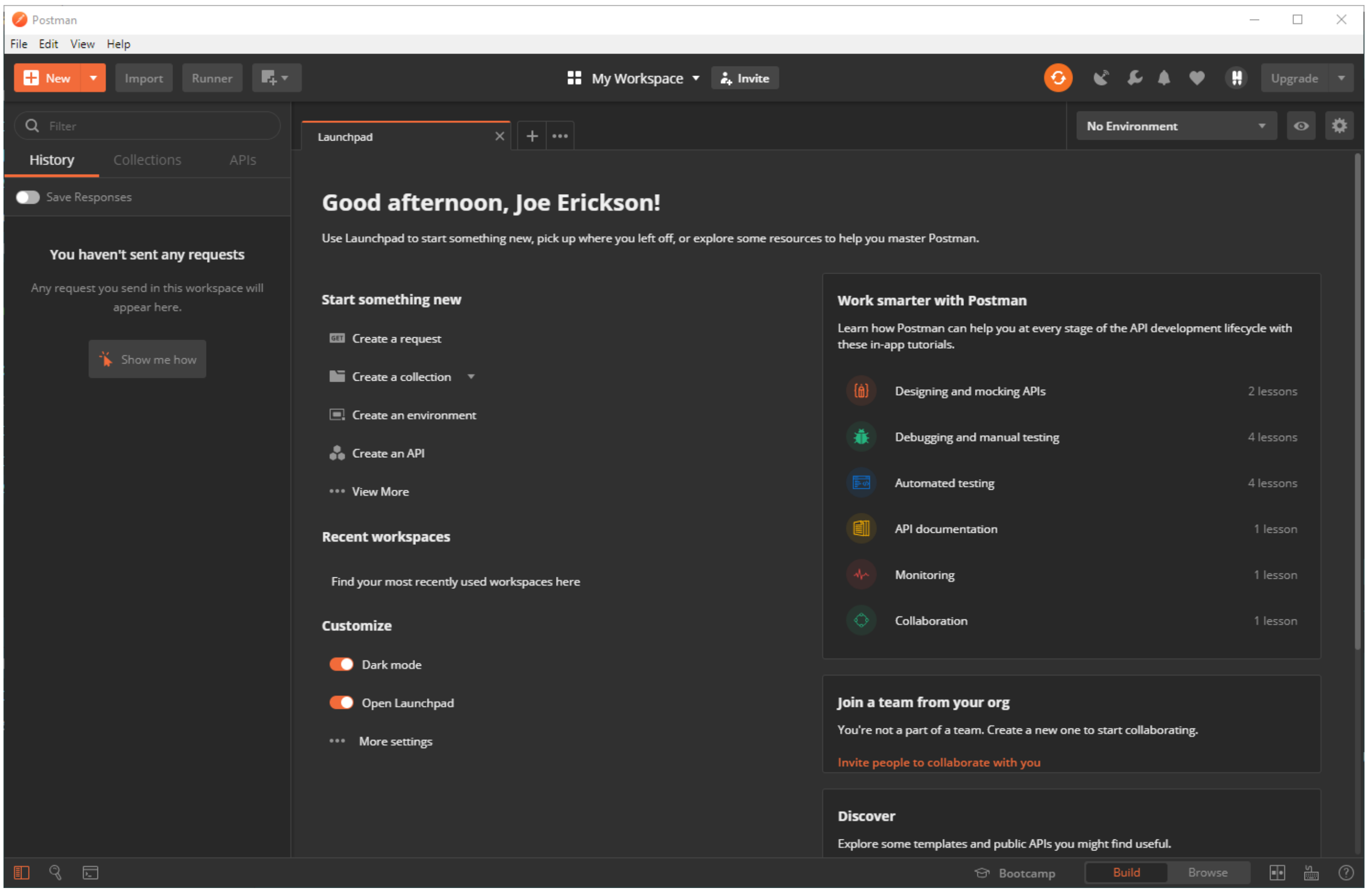
```
\{^_^}/ hi!

Loading ./db/instaounce.json
Done

Resources
http://localhost:3000/users
http://localhost:3000/images
http://localhost:3000/comments

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
```

If you see this message, it means that the web API server has started successfully and is now waiting for requests. To send it HTTP requests, you'll use Postman.

Step Two: Start Postman

Postman should be installed on your laptop already. The icon looks like this in either your start menu or in your Applications folder:

Double-click that icon. You should see a new screen like this:



You haven't sent any requests

Any request you send in this workspace will appear here.

Good afternoon, Joe Erickson!

Use Launchpad to start something new, pick up where you left off, or explore some resources to help you master Postman.

Find your most recently used workspaces here

Learn how Postman can help you at every stage of the API development lifecycle with these in-app tutorials.

You're not a part of a team. Create a new one to start collaborating.

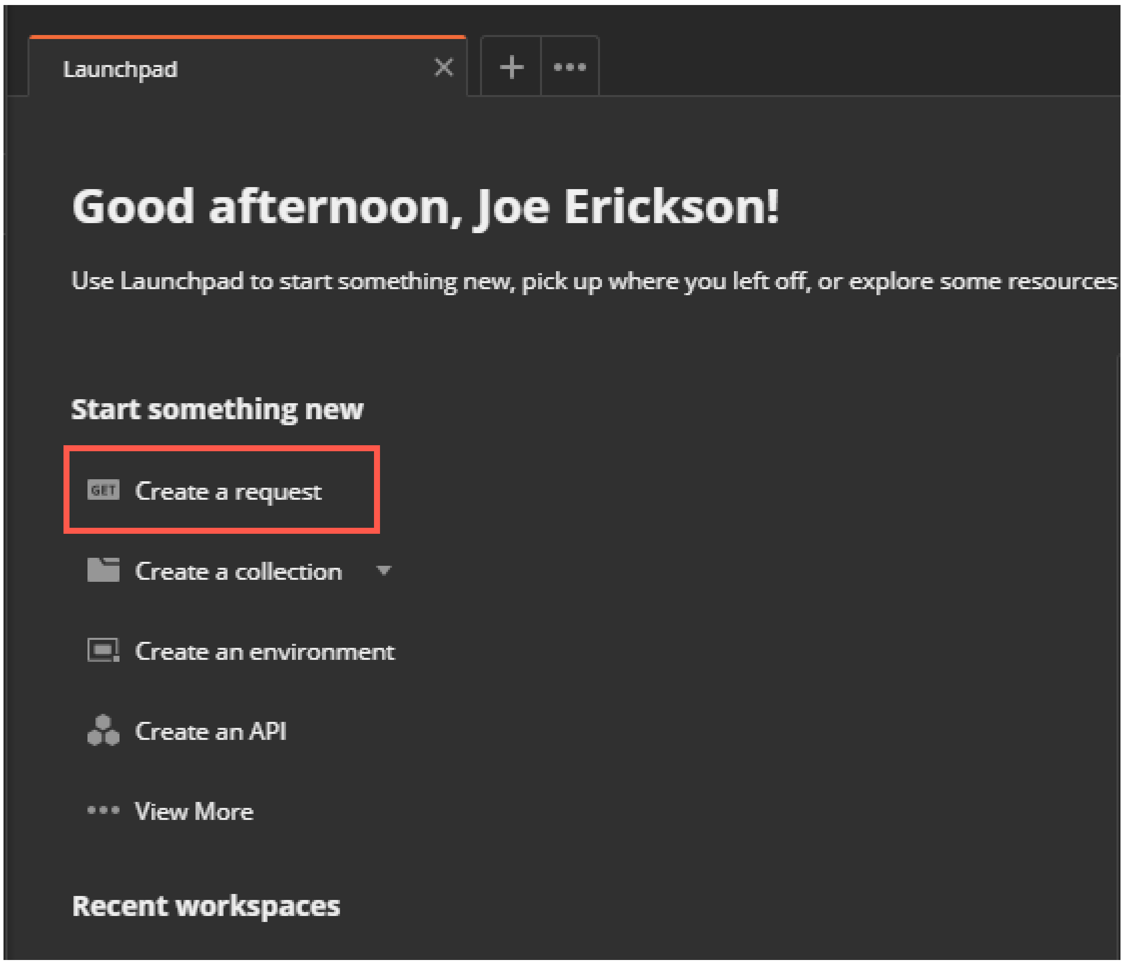
Explore some templates and public APIs you might find useful.

Postman is an application that lets you make HTTP requests like a browser but instead shows the raw information returned from the server. When you are interacting with a web server, that information is typically HTML. But when interacting with a web API, that information is usually JSON data, and a browser is not designed to display that information very well.

Web APIs also typically expect special HTTP headers and status codes that are more complex to set in a browser, so Postman is used in those circumstances. Think of it as your first interface to a Web API.

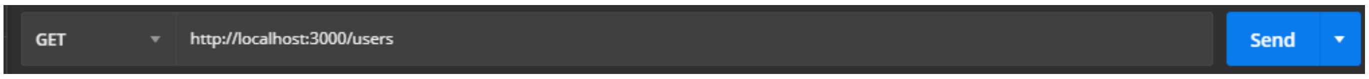
Step Three: Create a new request in Postman

In the Postman interface, click the "Create a request" link to start a new API session:



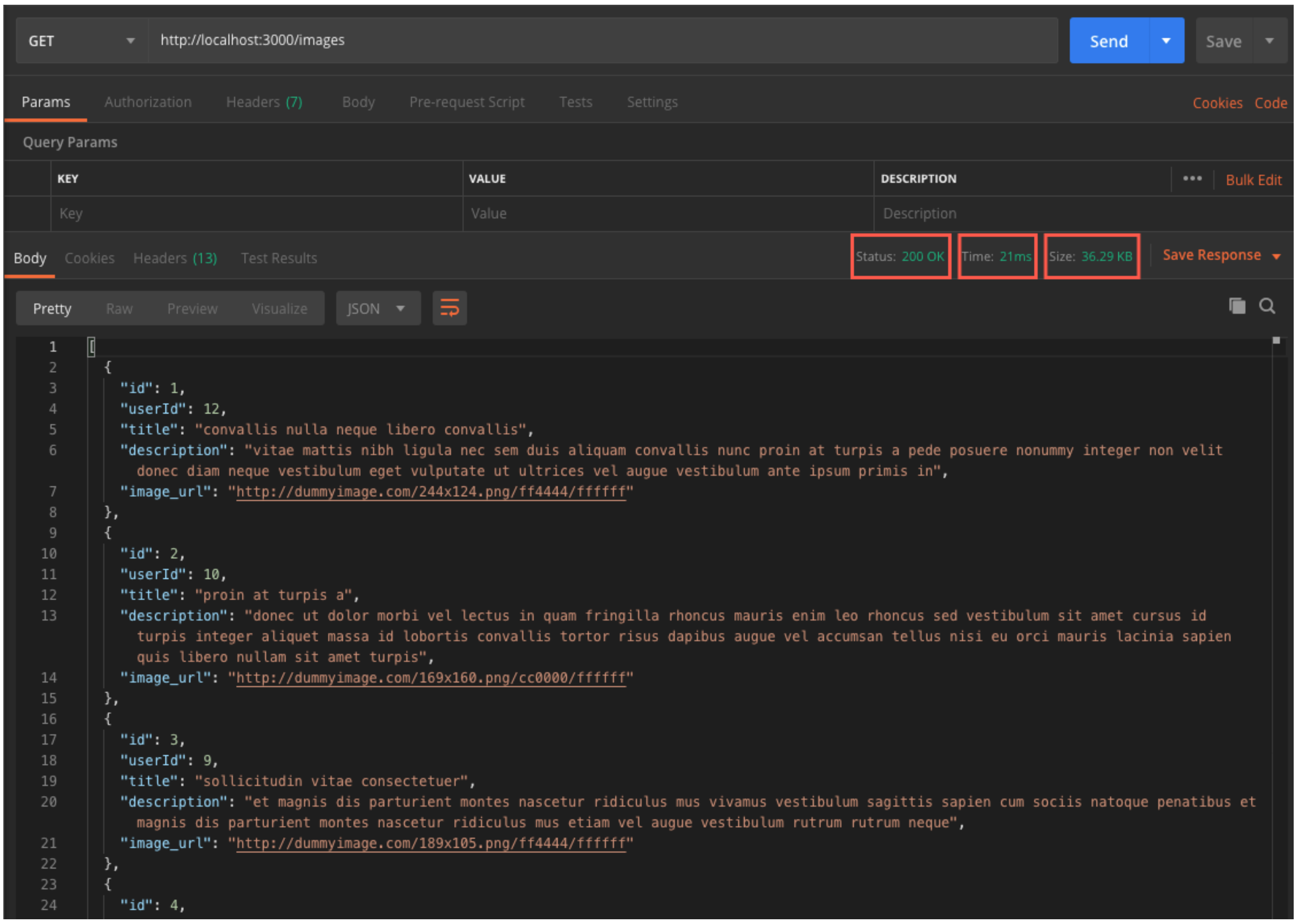
This session records and lets you play back API requests.

For your first request, make a request that returns all the users as an array. The URL for that request is `http://localhost:3000/users`. Enter the URL in the text box and then click send:



You should see underneath that the JSON data for 25 users has loaded in.

If you change the request to `http://localhost:3000/images`, you'll see the image information load in when you click Send:



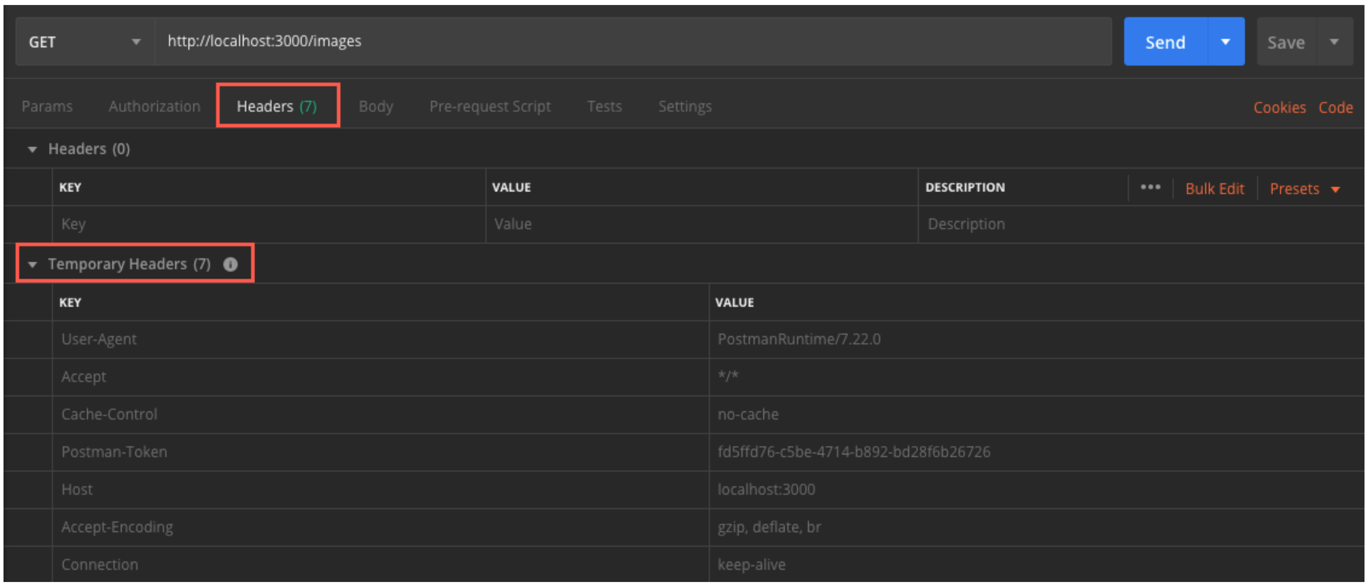
Status: 200 OK Time: 21ms Size: 36.29 KB

```
[
  {
    "id": 1,
    "userId": 12,
    "title": "convallis nulla neque libero convallis",
    "description": "vitae mattis nibh ligula nec sem duis aliquam convallis nunc proin at turpis a pede posuere nonummy integer non velit donec diam neque vestibulum eget vulputate ut ultrices vel augue vestibulum ante ipsum primis in",
    "image_url": "http://dummyimage.com/244x124.png/ff4444/ffffff"
  },
  {
    "id": 2,
    "userId": 10,
    "title": "proin at turpis a",
    "description": "donec ut dolor morbi vel lectus in quam fringilla rhoncus mauris enim leo rhoncus sed vestibulum sit amet cursus id turpis integer aliquet massa id lobortis convallis tortor risus dapibus augue vel accumsan tellus nisi eu orci mauris lacinia sapien quis libero nullam sit amet turpis",
    "image_url": "http://dummyimage.com/169x160.png/cc0000/ffffff"
  },
  {
    "id": 3,
    "userId": 9,
    "title": "sollicitudin vitae consectetuer",
    "description": "et magnis dis parturient montes nascetur ridiculus mus vivamus vestibulum sagittis sapien cum sociis natoque penatibus et magnis dis parturient montes nascetur ridiculus mus etiam vel augue vestibulum rutrum rutrum neque",
    "image_url": "http://dummyimage.com/189x105.png/ff4444/ffffff"
  },
  {
    "id": 4,
```

This shows the returned data. The Postman interface also shows you three important pieces of information about the request: the returned HTTP Status Code, how long the response took, and the size of the response data. You can use this information for debugging purposes—for example, how slow the API response is in case you need to compensate for it.

Step Four: Review request and response headers

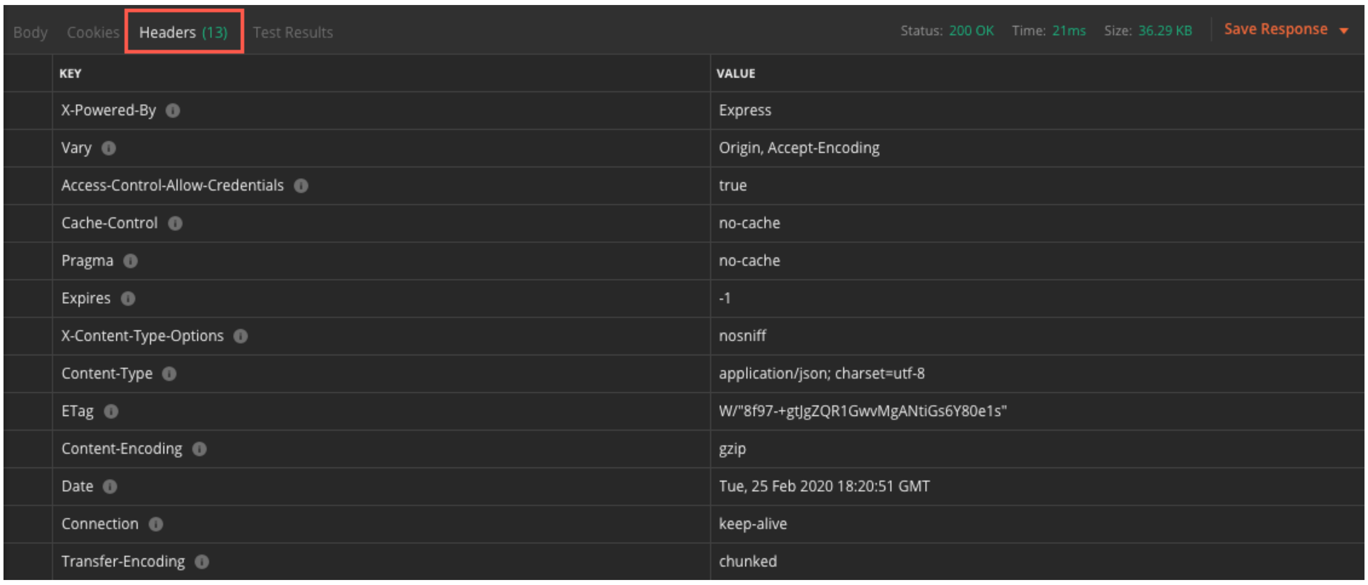
Clicking on Headers in the Request section lets you set HTTP headers for the request and see what HTTP headers Postman sets for the request:



KEY	VALUE
User-Agent	PostmanRuntime/7.22.0
Accept	*/*
Cache-Control	no-cache
Postman-Token	fd5ffd76-c5be-4714-b892-bd28f6b26726
Host	localhost:3000
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

Many of the temporary headers are headers that Postman sets by default, like the User-Agent and Host. These can be overwritten by your own headers for testing or exercising the back-end API.

You can also see the Response Headers that the server sends back in the Response view:



KEY	VALUE
X-Powered-By	Express
Vary	Origin, Accept-Encoding
Access-Control-Allow-Credentials	true
Cache-Control	no-cache
Pragma	no-cache
Expires	-1
X-Content-Type-Options	nosniff
Content-Type	application/json; charset=utf-8
ETag	W/"8f97-+gtJgZQR1GwvMgANtiGs6Y80e1s"
Content-Encoding	gzip
Date	Tue, 25 Feb 2020 18:20:51 GMT
Connection	keep-alive
Transfer-Encoding	chunked

These are the headers that the server sends back. These can be useful to look at if you need to debug the API.

Summary

You can now:

- Open Postman
- Make requests in Postman to web APIs
- View and set the HTTP Headers in the request
- View the Response Status Code, response time, size, and headers